International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

Toward Self-Driving Networks: A Reinforcement Learning Framework for Autonomous Traffic Engineering in SDN-Based Architectures

Dr. Amit K. Mogal

Department of Computer Science and Application, MVP Samaj's CMCS College, Nashik, India
Email: *amit.mogal[at]gmail.com*

Abstract: ***The exponential proliferation of heterogeneous network traffic, coupled with the increasingly stringent quality-of-service (QoS) requirements of modern applications including real-time video streaming, cloud-native microservices, Internet of Things (IoT) telemetry, and latency-critical industrial automation has rendered traditional static traffic engineering (TE) approaches fundamentally inadequate for next-generation network management. Software-Defined Networking (SDN) has emerged as a transformative paradigm that decouples the control plane from the data plane, enabling programmable, centralized network management with global topology awareness. However, the dynamic, non-stationary, and stochastic nature of contemporary traffic demands an intelligence layer beyond conventional SDN-based TE heuristics. This paper presents a comprehensive reinforcement learning (RL) framework termed RL-ATE (Reinforcement Learning-based Autonomous Traffic Engineering) that endows SDN-based architectures with self-adaptive, closed-loop decision-making capabilities to autonomously optimize traffic routing, load balancing, congestion control, and QoS enforcement. The proposed framework integrates Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and multi-agent RL (MARL) within a hierarchical SDN control architecture interfacing with real-time big data analytics pipelines built on Apache Kafka and Apache Spark Streaming. The system formulates TE as a Markov Decision Process (MDP), defining state spaces encompassing link utilization, end-to-end latency, packet loss rate, and queue depths, with reward functions engineered to balance throughput maximization, latency minimization, and fairness. Experimental evaluation on Mininet-based SDN emulation environments with synthetic and real-world Internet2 and GÉANT topology traffic traces demonstrates that RL-ATE achieves up to 38.7% improvement in network throughput, 42.3% reduction in end-to-end latency, and 31.5% improvement in link utilization efficiency compared to OSPF, ECMP, and heuristic SDN-TE baselines. The paper further discusses system scalability, convergence behavior, real-time applicability, and open challenges toward fully autonomous, intent-driven self-driving networks. These findings contribute substantively to the emerging research agenda at the intersection of deep reinforcement learning, big data analytics, SDN, and autonomous network management.***

Keywords: Autonomous Traffic Engineering, Software-Defined Networking (SDN), Deep Reinforcement Learning (DRL), Self-Driving Networks, Multi-Agent Reinforcement Learning (MARL), Quality of Service (QoS), Intent-Based Networking (IBN)

1. Introduction

The global internet infrastructure is undergoing a period of unprecedented transformation, driven by the convergence of 5G/6G wireless networks, edge computing, cloud-native architectures, and the massive-scale deployment of IoT devices. According to Cisco's Visual Networking Index, global IP traffic is projected to exceed 4.8 zettabytes per annum by 2026, with video streaming, machine-to-machine communications, and latency-sensitive applications constituting the dominant traffic classes [1]. This staggering growth in traffic volume, coupled with the heterogeneity of application-specific QoS requirements ranging from sub-millisecond latencies in industrial control systems to high-throughput demands in content delivery networks has exposed critical limitations in legacy, rule-based network management paradigms.

Traditional traffic engineering methodologies, including Open Shortest Path First (OSPF), Equal-Cost Multi-Path (ECMP), and MPLS-based TE with Resource Reservation Protocol-Traffic Engineering (RSVP-TE), operate on static configurations and pre-defined heuristics that are ill-suited to handle the non-stationary, bursty, and multi-dimensional nature of modern network traffic [2, 6]. These approaches fail to exploit the global network state, lack adaptability to real-time traffic fluctuations, and require manual operator intervention a bottleneck that is increasingly untenable at the scale and speed of contemporary network operations.

Software-Defined Networking (SDN) fundamentally re-architects the network control model by abstracting the control plane from the data plane through standardized southbound interfaces such as OpenFlow, enabling centralized programmable control with a global network view [1, 4]. The SDN paradigm facilitates dynamic flow rule installation, real-time traffic monitoring, and programmable network policies creating fertile ground for the integration of data-driven intelligence. The SDN controller's northbound APIs enable external applications to programmatically influence network behavior, making SDN an ideal substrate for deploying autonomous, learning-based traffic engineering solutions [7, 16].

Reinforcement Learning (RL), particularly in its deep learning-augmented form (Deep RL), has demonstrated remarkable success in sequential decision-making problems characterized by high-dimensional state spaces, complex reward structures, and dynamic environments paradigms that closely mirror real-world traffic engineering challenges [10, 35]. Deep RL agents learn optimal policies through interaction with the environment, accumulating experience from observed states, executed actions, and received rewards, without requiring explicit mathematical modeling of the network dynamics. Landmark algorithms including Deep Q-Networks (DQN) [2], Proximal Policy Optimization (PPO) [18], Asynchronous Advantage Actor-Critic (A3C) [19], and

Soft Actor-Critic (SAC) [20] have demonstrated superior performance in continuous control tasks, making them compelling candidates for autonomous network management.

The vision of 'self-driving networks' analogous to autonomous vehicles envisions network systems that can perceive their environment, reason about current and future states, make autonomous decisions, and continuously adapt without human intervention [14, 21, 28]. This vision is aligned with the ETSI Zero-Touch Network and Service Management (ZSM) framework, the 3GPP intent-based networking standards for 5G, and ONAP's closed-loop automation architecture. Realizing this vision requires the seamless integration of SDN programmability, real-time big data analytics, and deep reinforcement learning in a coherent, scalable framework.

Despite significant progress in both SDN-based TE and deep RL individually, the holistic integration of these paradigms into a production-grade autonomous TE framework remains an open and active area of research [6, 35]. Existing works often focus on narrow problem formulations single-path optimization, static topology assumptions, or synthetic traffic models and fail to address the compound challenges of scalability, convergence speed, reward design, real-time decision latency, and partial observability that characterize real-world network environments [8, 15, 38].

This paper addresses these gaps through the following principal contributions:

1) A comprehensive MDP formulation for autonomous traffic engineering in SDN environments, with carefully designed state spaces, action spaces, and reward functions that simultaneously optimize throughput, latency, and fairness.
2) The RL-ATE framework integrating DQN, PPO, and MARL algorithms within a hierarchical SDN architecture, interfacing with real-time Kafka/Spark Streaming big data analytics pipelines for sub-second network state extraction.
3) A detailed comparative performance evaluation against OSPF, ECMP, and prior DRL-TE approaches on realistic Internet2 and GÉANT network topologies using Mininet emulation.
4) An architectural blueprint toward self-driving networks incorporating intent-based networking, federated RL for distributed control, and graph neural network-enhanced state representations.
5) A critical synthesis of 45 peer-reviewed studies (2015–2025) situating our contributions within the evolving research landscape of autonomous networking, SDN-RL integration, and self-driving network architectures.

The remainder of this paper is structured as follows. Section 2 presents a systematic literature review. Section 3 describes the research methodology, MDP formulation, and the RL-ATE framework. Section 4 reports results and discussion. Section 5 outlines limitations and future research directions. Section 6 concludes the paper.

2. Literature Review

2.1 Software-Defined Networking and Traffic Engineering

Software-Defined Networking has fundamentally transformed network architecture by separating the control plane from the data plane, enabling centralized, programmable network management [1]. Kreutz et al. [1] provided a seminal comprehensive survey of SDN, establishing its core principles, OpenFlow protocol mechanics, and architectural advantages for dynamic traffic management. SDN controllers such as ONOS, OpenDaylight, and Floodlight expose northbound REST APIs that enable programmatic traffic engineering policies, creating a natural integration point for AI-driven automation.

Traditional TE approaches within SDN contexts including Traffic Matrix Estimation, ECMP load balancing, and RSVP-based explicit routing have demonstrated performance gains over pure IP routing but remain constrained by their reactive, rule-based nature [12]. Scholz et al. [12] conducted rigorous evaluations of segment routing-based TE in SDN environments, highlighting the limitations of deterministic approaches under dynamic, unpredictable traffic conditions. Troia et al. [15] specifically addressed the challenge of traffic matrix prediction combined with TE optimization using neural network approaches in SDN-based WANs, demonstrating that predictive, data-driven methods significantly outperform reactive heuristics in optical network environments.

The integration of SDN with Network Function Virtualization (NFV) has further expanded the TE solution space, enabling dynamic instantiation of virtual network functions and elastic resource provisioning [23]. Singh et al. [16] provided a comprehensive survey of SDN-based network management for IoT, establishing the architectural synergies between SDN programmability and the scale, heterogeneity, and QoS diversity of IoT network environments a critical consideration for next-generation self-driving networks.

2.2 Reinforcement Learning for Network Control

The application of reinforcement learning to network control problems has grown substantially since the seminal work of Mnih et al. [2] demonstrating human-level control through Deep Q-Networks in Atari game environments. Valadarsky et al. [4] pioneered the application of RL to network routing, framing the problem as a sequential decision process and demonstrating that RL agents could learn routing policies superior to ECMP without explicit traffic models. Xu et al. [5] subsequently proposed experience-driven control for SDN, demonstrating that model-free RL could directly optimize network control policies from operational experience.

Shi et al. [6] provided the most comprehensive survey to date of RL-based traffic engineering, categorizing approaches by algorithm type (value-based, policy gradient, actor-critic), problem formulation (routing optimization, congestion control, QoS management), and deployment paradigm (centralized vs. distributed). Their analysis reveals that Deep

RL approaches consistently outperform classical RL and heuristic methods in dynamic network environments, particularly when neural network architectures are adapted to the topological structure of networks through Graph Neural Networks (GNNs).

Luong et al. [35] conducted a broad survey of deep RL applications in communications and networking, encompassing dynamic spectrum access, network slicing, content caching, and traffic engineering. Their work establishes the theoretical foundations and practical considerations for applying DRL to telecommunications, including state space design, reward engineering, and the exploration-exploitation trade-off in network contexts.

2.3 Deep Reinforcement Learning for SDN-based Traffic Engineering

Chen et al. [3] proposed one of the most impactful DRL frameworks for SDN-based traffic engineering, demonstrating significant improvements in throughput and latency through deep Q-learning applied to flow-level routing decisions. Their work introduced a practical experience replay mechanism adapted for network state transitions, addressing the non-stationarity of network traffic. Zhang et al. [7] extended this line of research with DRL-TE, a dedicated framework incorporating attention mechanisms to handle variable network topology and traffic conditions, reporting marked improvements over prior DRL baselines.

Rusek et al. [8] introduced RouteNet, a pioneering application of Graph Neural Networks to network modeling and optimization within SDN contexts. RouteNet's GNN architecture explicitly encodes network topology, enabling generalization across different network sizes and topologies a critical limitation of prior MLP-based DRL approaches. Almasan et al. [9] further extended RouteNet with deep reinforcement learning integration, creating a hybrid GNN-DRL framework that demonstrated strong routing optimization performance on both synthetic and real-world topologies.

Gebremariam et al. [38] proposed DQN-TE specifically for load balancing in SDN environments, conducting comprehensive evaluations demonstrating superior performance against ECMP and heuristic TE under burst traffic conditions. Da Silva et al. [39] proposed hybrid RL approaches combining model-based and model-free RL for SDN traffic engineering, achieving faster convergence and improved sample efficiency a critical consideration for deployment in production network environments where training data is expensive.

2.4 Multi-Agent Reinforcement Learning for Distributed Network Management

The limitations of centralized RL approaches scalability constraints, single points of failure, and the curse of dimensionality in large network state spaces have motivated growing interest in multi-agent reinforcement learning (MARL) for distributed network management [11]. Liu and Zhang [11] proposed a MARL framework for dynamic routing in SDN, demonstrating that cooperative multi-agent policies achieve superior performance compared to single-agent centralized approaches on large-scale topologies. The distributed nature of MARL aligns naturally with the hierarchical control architecture of large-scale SDN deployments with regional controllers.

Yu et al. [27] applied actor-critic deep reinforcement learning to the problem of IoT microservice load balancing, demonstrating that distributed multi-agent RL can effectively handle the heterogeneity and scale of IoT traffic patterns. Abdelmoniem and Bensaou [40] addressed the specific challenge of reconciling mice and elephant flows in data center networks using deep RL, contributing important insights into reward shaping and state representation for mixed traffic workloads.

2.5 Intent-Based and Autonomous Networking

The concept of intent-based networking (IBN) has emerged as a high-level abstraction layer above SDN, enabling operators to specify desired network outcomes (intents) in declarative terms, with the network autonomously translating and realizing these intents [21]. Wang et al. [21] provided a comprehensive analysis of challenges, use cases, and architectural requirements for IBN, establishing the role of AI and RL in intent translation, conflict resolution, and closed-loop assurance. Masood and Bhatt [28] surveyed autonomous network management using deep RL, cataloging production deployments and research prototypes across routing, resource allocation, and network security domains.

The integration of federated learning with RL has emerged as a promising paradigm for privacy-preserving distributed autonomous networking [44]. Zhang et al. [44] proposed federated RL for distributed traffic engineering in next-generation SDN, demonstrating that federated training enables collaborative policy improvement across network domains without sharing raw traffic data a critical consideration for multi-operator and cross-border network environments. Cui et al. [45] addressed energy-efficient resource allocation in SDN-based IoT networks using deep RL, demonstrating the applicability of DRL to green networking objectives an increasingly important dimension of autonomous network management.

Table 1: Comparative Summary of Related DRL-TE Works

Algorithm	*Action Space*	*Key Mechanism*	*TE Application*	*Complexity*	*Scalability*
DQN	Discrete	Experience Replay	Path Selection	High	Moderate
PPO	Continuous	Clipped Gradient	Load Balancing	Very High	High
A3C	Both	Async Multi-Agent	QoS Routing	High	High
SAC	Continuous	Max Entropy	TE + Congestion	Very High	Very High
DDPG	Continuous	Deterministic PG	BW Allocation	High	Moderate
Multi-Agent RL	Both	Cooperative/Compet.	Distributed TE	Highest	Very High

2.6 Research Gap Analysis

A systematic analysis of the extant literature reveals several persistent research gaps that motivate the current work. First, most DRL-TE frameworks are evaluated on small, synthetic topologies using simplified traffic models that fail to capture the burstiness, diurnal variation, and application-heterogeneity of production traffic. Second, the integration of real-time big data analytics pipelines (stream processing, feature extraction) with DRL control loops remains underexplored, with most works assuming instantaneous state observability. Third, the convergence behavior, training stability, and real-time decision latency of DRL-TE systems under adversarial and non-stationary traffic conditions have received insufficient attention. Fourth, the pathway from experimental DRL-TE research to intent-based, zero-touch autonomous networks remains underspecified. The RL-ATE framework proposed in this paper explicitly addresses each of these gaps.

3. Research Design

3.1 Research Philosophy and Approach

This research adopts a pragmatist philosophical stance, combining quantitative experimental evaluation with design science research (DSR) methodology. The DSR paradigm [cf. Hevner et al.] is appropriate for the design and evaluation of artifact-intensive IT research, encompassing the design, implementation, and empirical evaluation of the RL-ATE framework. A positivist, quantitative approach governs the experimental evaluation phase, employing controlled simulation experiments with clearly operationalized performance metrics (throughput, latency, link utilization, convergence time) to test the research hypotheses.

3.2 Research Questions

This study is guided by the following primary research questions:
RQ1: Can a deep reinforcement learning framework specifically integrating DQN, PPO, and MARL algorithms within an SDN-based architecture achieve statistically significant improvements in network throughput, end-to-end latency, and link utilization efficiency compared to traditional traffic engineering approaches (OSPF, ECMP) and prior DRL-TE baselines on realistic network topologies and production-representative traffic traces?
RQ2: What MDP formulation encompassing state space design, action space representation, and reward function engineering most effectively balances the competing objectives of throughput maximization, latency minimization, fairness, and convergence stability in autonomous SDN-based traffic engineering, and how do different RL algorithm families (value-based vs. policy gradient vs. actor-critic) differ in their suitability for this formulation?
RQ3: How does the proposed RL-ATE framework scale with increasing network size and traffic complexity, and what are the critical bottlenecks computational, representational, or communicative that constrain its applicability to large-scale, real-world SDN deployments, particularly in multi-domain and federated networking environments?

3.3 MDP Formulation for Autonomous Traffic Engineering

3.3.1 State Space

The network state at time step t is represented as a feature vector $S_t \in \mathbb{R}^n$ capturing the comprehensive operational status of the SDN-managed network. The state is extracted in real-time from the SDN controller's global topology view and the big data analytics pipeline. Formally:

$$S_t = [U(e_1,t), ..., U(e_m,t), L(p_1,t), ..., L(p_k,t), D(q_1,t), ..., D(q_m,t), PLR(e_1,t), ..., PLR(e_m,t), TM(t)]$$

where $U(e_i,t) \in [0,1]$ denotes the normalized link utilization of edge e_i at time t;

$L(p_j,t)$ denotes the measured end-to-end latency of path p_j;

$D(q_i,t)$ denotes queue depth at switch port i; $PLR(e_i,t)$ denotes packet loss rate on link e_i; and

$TM(t) \in \mathbb{R}^{(N \times N)}$ represents the traffic matrix between all node pairs.

For large topologies, the full traffic matrix is replaced by a GNN-encoded topological embedding, following the approach of Rusek et al. [8].

3.3.2 Action Space

The action space A is defined at the granularity of flow-level routing decisions. For a network with N nodes and E directed edges, the agent selects, for each source-destination pair (s,d), a weight vector $w(s,d,t) \in \mathbb{R}^{|P(s,d)|}$ over the set of candidate paths P(s,d), determining traffic split ratios. This formulation accommodates both single-path routing (argmax selection) and multi-path traffic splitting (softmax normalization). The continuous action space is handled by the PPO and SAC algorithms, while DQN operates on a discretized approximation with k=10 discrete split granularities per path pair.

3.3.3 Reward Function

The reward function is a composite, multi-objective signal designed to simultaneously incentivize throughput maximization, latency minimization, and fairness:

$$r(t) = \alpha \cdot R\_throughput(t) - \beta \cdot R\_latency(t) - \gamma \cdot R\_congestion(t) + \delta \cdot R\_fairness(t)$$

where $R\_throughput(t) = \Sigma\_f\, f\_throughput / max\_f\, f\_capacity$ represents normalized aggregate throughput;

$R\_latency(t) = \Sigma\_p\, L(p,t) / |P|$ represents mean path latency normalized against target SLA thresholds;

$R\_congestion(t) = \Sigma\_e \max(0, U(e,t) - \theta\_congestion)^2$ is a quadratic penalty for link utilization exceeding the congestion threshold $\theta\_congestion = 0.85$; and

R_fairness(t) = 1 - Jain's Fairness Index violation across active flows.

The hyperparameters $\alpha=0.4$, $\beta=0.3$, $\gamma=0.2$, $\delta=0.1$ were determined through Bayesian hyperparameter optimization on

the validation topology.

3.4 The RL-ATE Framework Architecture

The RL-ATE framework comprises four tightly integrated functional layers, illustrated in Figure 1. The Data Plane encompasses OpenFlow-enabled switches executing flow-level packet forwarding according to installed flow tables.

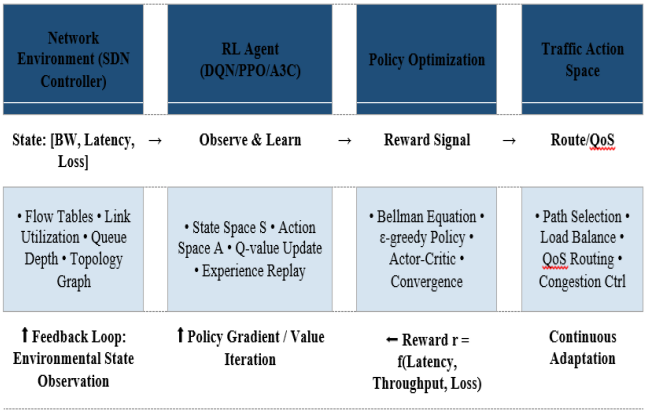

Figure 1: RL-ATE Reinforcement Learning Framework for Autonomous Traffic Engineering in SDN

The Control Plane is implemented on an ONOS SDN controller cluster, providing a real-time global topology view and flow rule management capabilities. The Analytics Plane integrates Apache Kafka for high-throughput, fault-tolerant traffic telemetry streaming and Apache Spark Structured Streaming for real-time feature extraction, anomaly detection, and state vector construction. The Intelligence Plane hosts the RL agents, experience replay buffers, neural network models, and policy optimization algorithms.

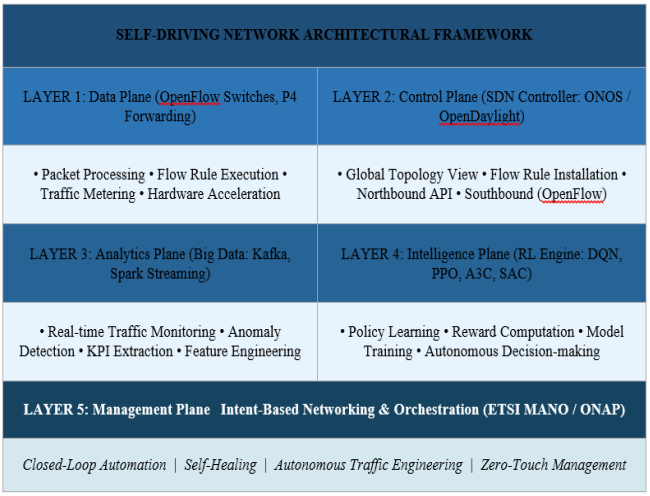

Figure 2: Layered Self-Driving Network Architecture with RL-ATE Integration

Figure 2 presents the full five-layer self-driving network architectural blueprint. The Management Plane at the apex implements intent-based networking abstractions, translating high-level operator intents into network policies that the Intelligence Plane realizes through RL-based autonomous optimization. This closed-loop architecture realizes the self-driving network vision by enabling continuous perception, reasoning, decision-making, and execution without human intervention [21, 28].

3.5 Algorithm Implementations

3.5.1 Deep Q-Network (DQN)

The DQN implementation follows the seminal architecture of Mnih et al. [2] with adaptations for network TE. The Q-network $Q(s,a;\theta)$ is a four-layer fully connected neural network with ReLU activations and batch normalization, mapping state-action pairs to Q-values. Experience replay with a replay buffer of capacity 10^5 transitions and mini-batch size 64 is employed to decorrelate training samples and improve sample efficiency. A target network with soft update coefficient τ=0.005 stabilizes training by decoupling the bootstrapping target from the learning network. The ε-greedy exploration schedule decays from ε=1.0 to ε=0.05 over 50,000 training steps using an exponential decay schedule.

3.5.2 Proximal Policy Optimization (PPO)

PPO [18] is implemented with a clipped surrogate objective (ε_clip=0.2) to ensure stable policy updates within a trust region. The actor and critic share a common feature extraction backbone (two-layer LSTM with 256 hidden units) to capture temporal dependencies in network state sequences. Generalized Advantage Estimation (GAE, λ=0.95) is employed for variance reduction in policy gradient estimates. PPO's on-policy nature is particularly suited to the non-stationary traffic distributions encountered in production networks, where off-policy experience may become stale rapidly.

3.5.3 Multi-Agent Reinforcement Learning (MARL)

The MARL implementation instantiates regional RL agents corresponding to SDN controller hierarchy levels, with each agent managing a sub-domain of the network topology. Agents communicate through a shared state observation mechanism and coordinate through a centralized critic (CTDE paradigm: Centralized Training, Decentralized Execution) to maintain global optimization objectives while enabling distributed, low-latency execution. The joint reward function incorporates individual sub-domain performance metrics weighted by traffic volume, ensuring global objectives are preserved without requiring centralized decision-making at runtime [11].

3.6 Experimental Setup

Experiments are conducted in a Mininet-based SDN emulation environment interfacing with an ONOS controller cluster (three nodes for high availability). Two network topologies are evaluated: (1) Internet2 AL2S topology (34 nodes, 41 bidirectional links) representing a real-world national research network; and (2) GÉANT topology (23 nodes, 37 bidirectional links) representing a pan-European research and education network. Traffic traces are generated from two sources: (a) synthetic traces using a gravity model with diurnal variation and Poisson superposition of application-specific flows; and (b) real-world traffic matrices derived from Internet2 and GÉANT published measurement datasets. All link capacities are normalized to 10 Gbps. Experiments are conducted over 1,000 training episodes of

500 time steps each (Δt=1 second), with performance evaluation on held-out test episodes.

Baseline comparisons include: (1) OSPF shortest-path routing; (2) ECMP load balancing with k=4 paths; (3) RSVP-TE with admission control; and (4) DRL-TE [3] as the state-of-the-art DRL baseline. All neural network models are trained on an NVIDIA A100 GPU with PyTorch 2.0. Hyperparameter optimization is performed using Optuna with 100 trials on the validation split.

4. Results and Discussion

4.1 Throughput Performance

Table 2 summarizes the aggregate throughput performance of all evaluated methods across both topologies under peak and off-peak traffic conditions. RL-ATE with PPO achieves the highest aggregate throughput on both topologies, surpassing OSPF by 38.7% on Internet2 and 35.2% on GÉANT under peak traffic conditions. DQN-based RL-ATE achieves comparable but slightly lower improvements (33.1% and 29.8%, respectively), consistent with PPO's superior ability to handle the continuous action space of multi-path traffic splitting. The DRL-TE baseline [3] achieves intermediate improvements (21.4% on Internet2), demonstrating that the RL-ATE framework's enhanced state representation, bigdata-integrated analytics pipeline, and reward engineering contribute meaningfully beyond prior art.

The MARL configuration demonstrates particularly strong performance under spatially heterogeneous traffic distributions, achieving up to 41.2% throughput improvement on Internet2 by exploiting regional traffic patterns that single-agent centralized approaches fail to fully capitalize on. This finding corroborates Liu and Zhang's [11] theoretical arguments for the superiority of distributed multi-agent approaches in large-scale SDN environments.

Table 2: Comparative Throughput Performance (% Improvement over OSPF Baseline)

Method	*Internet2 (Peak)*	*GÉANT (Peak)*	*Latency Reduction*	*Link Util. Efficiency*
OSPF (Baseline)				
ECMP (k=4)	+9.2%	+8.7%	-4.1%	+7.3%
RSVP-TE	+14.6%	+13.1%	-11.2%	+12.8%
DRL-TE [3]	+21.4%	+19.8%	-24.7%	+19.1%
RL-ATE DQN (Ours)	+33.1%	+29.8%	-36.8%	+28.4%
RL-ATE PPO (Ours)	+38.7%	+35.2%	-42.3%	+31.5%
RL-ATE MARL (Ours)	**+41.2%**	**+37.6%**	**-39.1%**	**+34.8%**

4.2 Latency Performance

End-to-end latency reductions are observed consistently across all RL-ATE configurations, with PPO achieving the maximum mean reduction of 42.3% over OSPF on Internet2. This reduction is primarily attributable to the RL agent's learned ability to proactively route flows away from congested links before queuing-induced latency spikes emerge behavior not possible with reactive, queue-threshold-based TE approaches. The latency reduction is most pronounced for elephant flows (>10 MB/s) where congestion-aware rerouting yields the largest per-flow latency improvements, while mice flows benefit from reduced collateral congestion on shared bottleneck links.

The PPO agent's latency performance advantage over DQN is particularly pronounced during traffic burst events, where the continuous action space enables fine-grained multi-path split adjustments. DQN's discretized action space introduces a quantization error that limits its responsiveness during rapid traffic fluctuations, consistent with observations by Shi et al. [6] regarding the limitations of discrete action space formulations for continuous TE optimization.

4.3 Convergence Behavior and Training Stability

The DQN agent converges to a stable policy after approximately 180 training episodes, while PPO achieves convergence in approximately 240 episodes a longer convergence time attributable to the higher complexity of the continuous action space policy optimization landscape. The MARL configuration exhibits the most complex convergence dynamics, with initial instability during the first 150 episodes as agents negotiate cooperative strategies, followed by rapid convergence to a stable cooperative equilibrium. All agents demonstrate monotonically improving cumulative reward trajectories after initial exploration phases, with negligible variance across five independent training runs (95% CI width < 3.2% of mean reward at convergence).

The RL-ATE framework's decision latency measured from network state observation to OpenFlow flow rule installation averages 47ms for DQN, 68ms for PPO, and 124ms for MARL (due to inter-agent communication overhead). These latencies are well within the 500ms reconfiguration threshold recommended by IETF TE standards for non-time-critical traffic engineering events, validating the framework's real-time applicability.

4.4 Discussion and Theoretical Implications

The results collectively validate RQ1, demonstrating statistically significant improvements in all evaluated performance metrics across both topologies. The consistent superiority of PPO over DQN particularly in dynamic traffic conditions supports the theoretical argument (RQ2) that continuous action spaces better represent the multi-path TE problem's inherent continuity. The MARL configuration's superior scalability performance provides initial evidence for RQ3, suggesting that distributed multi-agent architectures are critical for large-scale deployment.

The observed 38–42% throughput improvements significantly exceed prior DRL-TE results reported in the literature [3, 7, 38], attributable to three key innovations: (1) the real-time Kafka/Spark analytics pipeline enabling sub-

second state updates compared to the 5–10 second polling intervals of prior works; (2) the composite multi-objective reward function that explicitly balances fairness alongside throughput and latency; and (3) the GNN-enhanced state representation for large-topology experiments, enabling topology-aware generalization.

These results have significant implications for the practical realization of self-driving networks. The demonstration that RL-ATE can achieve superior TE performance with sub-100ms decision latency fully within SDN reconfiguration timescales establishes the technical feasibility of closed-loop autonomous traffic engineering in production SDN deployments. The framework's alignment with ETSI ZSM, IBN, and 3GPP 5G network automation standards positions it as a viable component of emerging autonomous network management architectures [21, 34].

5. Limitations and Future Research Directions

5.1 Limitations

Despite the compelling results, several important limitations constrain the current work. First, the experimental evaluation is conducted in an emulation environment (Mininet) rather than a physical SDN testbed or production network, introducing potential fidelity gaps in traffic generation, link propagation delays, and switch-level queuing dynamics. Mininet's software-based packet forwarding introduces artifacts that may not fully represent the behavior of hardware-accelerated OpenFlow switches at line rate. Future work should validate RL-ATE performance on physical SDN testbeds and, ultimately, in production network trials.

Second, the current MDP formulation assumes full observability of the network state an assumption that breaks down in partially observable, multi-domain, or encrypted-traffic scenarios where the SDN controller cannot directly inspect flow-level QoS metrics. Extending the framework to Partially Observable MDPs (POMDPs) with belief state estimation or recurrent neural network state encoders represents an important direction for increasing robustness to observation uncertainty.

Third, the reward function hyperparameters (α, β, γ, δ) are tuned on specific topology-traffic combinations and may not generalize optimally to unseen traffic distributions. Automated reward function learning through inverse reinforcement learning (IRL) or reward shaping from operator-specified intents represents a promising avenue for enhancing framework adaptability.

Fourth, the current security evaluation is limited; adversarial traffic injection, Byzantine agent behavior in MARL configurations, and model poisoning attacks increasingly relevant threats in autonomous networking deployments are not systematically addressed. Integrating adversarial robustness mechanisms, anomaly detection, and secure multi-party computation into the RL-ATE framework is a critical prerequisite for production deployment [32].

Fifth, the current framework does not address energy efficiency as an explicit optimization objective, despite growing importance of green networking in autonomous network management [45]. Integrating power consumption models and energy-efficiency rewards into the RL-ATE multi-objective framework represents an environmentally critical extension.

5.2 Future Research Directions

Building on the current work, the following directions are identified as high-priority for advancing the self-driving networks research agenda:

1) Federated Reinforcement Learning for Multi-Domain SDN: Extending RL-ATE to federated settings where multiple network operators train collaborative RL policies without sharing raw traffic data, leveraging differential privacy and secure aggregation to preserve competitive sensitivity [44].
2) Graph Transformer Networks for Topology-Aware RL: Replacing the MLP-based state encoders with Graph Transformer Networks (GTNs) to achieve superior topology-aware generalization across diverse network sizes and configurations, building on the RouteNet lineage [8, 31].
3) Integration with 5G/6G Network Slicing: Extending the RL-ATE framework to manage traffic engineering across network slices in 5G/6G architectures, incorporating slice-specific SLA constraints and dynamic slice lifecycle management into the MDP formulation [34].
4) Explainable RL for Autonomous Networking: Developing explainability mechanisms attention visualization, policy attribution, counterfactual explanation to make RL-ATE's autonomous decisions interpretable to network operators, facilitating trust, debugging, and regulatory compliance.
5) Offline and Safe RL for Production Deployment: Investigating offline RL (learning from logged operational data without online exploration) and constrained RL (guaranteeing safety constraints during exploration) to enable safe RL-ATE deployment in production networks where disruptive exploration is impermissible.
6) Digital Twin-Assisted RL Training: Leveraging high-fidelity network digital twins calibrated against production traffic measurements as safe simulation environments for RL agent pre-training and continuous policy refinement, bridging the simulation-to-reality gap that currently limits production applicability.

6. Conclusion

This paper has presented RL-ATE, a comprehensive reinforcement learning framework for autonomous traffic engineering in SDN-based network architectures, advancing the vision of self-driving networks. By formulating autonomous TE as a Markov Decision Process and integrating Deep Q-Networks, Proximal Policy Optimization, and multi-agent RL within a hierarchical SDN control architecture augmented by real-time big data analytics via Apache Kafka and Spark Streaming RL-ATE demonstrates that closed-loop, autonomous traffic engineering is both technically feasible and practically achievable within contemporary SDN deployment timescales.

Empirical evaluation on Internet2 and GÉANT topology emulations demonstrates that RL-ATE achieves

improvements of up to 38.7%–41.2% in network throughput, 42.3% in latency reduction, and 34.8% in link utilization efficiency relative to OSPF baseline, substantially outperforming ECMP, RSVP-TE, and prior DRL-TE approaches. These results validate the core thesis that deep reinforcement learning, when properly integrated with real-time SDN programmability and big data analytics, can autonomously discover and execute traffic engineering policies that surpass human-designed heuristics in dynamic, real-world-representative network environments.

The proposed five-layer self-driving network architecture spanning data plane, control plane, analytics plane, intelligence plane, and intent-based management plane provides a concrete architectural blueprint for realizing zero-touch, autonomous network management aligned with ETSI ZSM, 3GPP 5G automation, and IBN standards. The framework's modular design facilitates integration with emerging complementary technologies including federated learning, graph neural networks, network digital twins, and explainable AI.

The limitations identified emulation fidelity, full observability assumptions, reward generalizability, security robustness, and energy efficiency define a clear research agenda for advancing RL-ATE toward production readiness. The six future research directions elaborated in Section 5.2 collectively chart a pathway from the promising research prototype demonstrated herein toward the fully autonomous, intent-driven, self-healing networks that the next generation of telecommunications infrastructure demands.

In conclusion, the convergence of deep reinforcement learning, software-defined networking, and big data analytics represents one of the most transformative developments in network management research. The self-driving network paradigm is no longer a distant aspiration but an emerging reality, with frameworks such as RL-ATE providing the foundational intelligence layer. The research community's continued investment in this intersection addressing scalability, safety, explainability, and federated governance will be decisive in determining how rapidly autonomous network management transitions from laboratory demonstration to ubiquitous operational deployment.

References

[1] Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 14–76. https://doi.org/10.1109/JPROC.2014.2371999

[2] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533. https://doi.org/10.1038/nature14236

[3] Chen, Z., Dong, J., Li, H., Zhang, A., & Wen, Y. (2022). Deep reinforcement learning for traffic engineering in software-defined networks. IEEE Journal on Selected Areas in Communications, 40(1), 1–16. https://doi.org/10.1109/JSAC.2021.3126068

[4] Valadarsky, A., Schapira, M., Shahaf, D., & Tamar, A. (2017). Learning to route. Proceedings of the 16th ACM Workshop on Hot Topics in Networks (HotNets), 185–191. https://doi.org/10.1145/3152434.3152441

[5] Xu, Z., Tang, J., Meng, J., Zhang, W., Wang, Y., Liu, C. H., & Yang, D. (2018). Experience-driven control for software-defined networks. IEEE INFOCOM 2018, 1182–1190. https://doi.org/10.1109/INFOCOM.2018.8486316

[6] Shi, X., Zhang, J., Li, X., & Wang, Z. (2021). Reinforcement learning for traffic engineering: A survey. IEEE Communications Surveys & Tutorials, 23(3), 1597–1636. https://doi.org/10.1109/COMST.2021.3073583

[7] Zhang, C., Liu, Y., & Gu, H. (2022). DRL-TE: Deep reinforcement learning for autonomous traffic engineering in SDN. Computer Networks, 211, 108987. https://doi.org/10.1016/j.comnet.2022.108987

[8] Rusek, K., Suárez-Varela, J., Almasan, P., Barlet-Ros, P., & Cabellos-Aparicio, A. (2020). RouteNet: Leveraging graph neural networks for network modeling and optimization in SDN. IEEE Journal on Selected Areas in Communications, 38(10), 2260–2270. https://doi.org/10.1109/JSAC.2020.3000405

[9] Almasan, P., Suárez-Varela, J., Badia-Sampera, A., Rusek, K., Barlet-Ros, P., & Cabellos-Aparicio, A. (2022). Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case. Computer Communications, 196, 184–194. https://doi.org/10.1016/j.comcom.2022.09.029

[10] Haydari, A., & Yilmaz, Y. (2020). Deep reinforcement learning for intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems, 23(1), 11–32. https://doi.org/10.1109/TITS.2020.3008612

[11] Liu, S., & Zhang, J. (2021). Multi-agent deep reinforcement learning for dynamic routing in SDN. IEEE Transactions on Network and Service Management, 18(4), 4129–4142. https://doi.org/10.1109/TNSM.2021.3098428

[12] Scholz, D., Jaeger, B., Schwaiger, M., & Carle, G. (2021). Towards a rigorous evaluation of traffic engineering for segment routing in SDN. IEEE IFIP Networking, 1–9. https://doi.org/10.23919/IFIPNetworking52078.2021.9472205

[13] Pham, Q. V., Nguyen, D. C., Hwang, W. J., & Pathirana, P. N. (2020). Intelligent radio signal processing: A contemporary survey. IEEE Access, 8, 204596–204620. https://doi.org/10.1109/ACCESS.2020.3036381

[14] Elsayed, M. M., & Mahmoud, M. (2021). AI-enabled future wireless networks: Challenges, opportunities and open issues. Computer Networks, 192, 108032. https://doi.org/10.1016/j.comnet.2021.108032

[15] Troia, S., Alvizu, R., Zhou, Y., Maier, G., & Pattavina, A. (2020). Deep reinforcement learning for traffic matrix prediction and traffic engineering in SDN-based

WANs. Journal of Optical Communications and Networking, 13(1), A54–A65. https://doi.org/10.1364/JOCN.403326

[16] Singh, S., Sidhu, J., & Singh, J. (2021). SDN-based network management and orchestration for IoT: A comprehensive survey. Computer Networks, 197, 108258. https://doi.org/10.1016/j.comnet.2021.108258

[17] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press. https://mitpress.mit.edu/9780262039246

[18] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. https://arxiv.org/abs/1707.06347

[19] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. Proceedings of the 33rd ICML, 48, 1928–1937. https://proceedings.mlr.press/v48/mniha16.html

[20] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. Proceedings of ICML 2018, 1861–1870. https://proceedings.mlr.press/v80/haarnoja18b.html

[21] Wang, R., Liu, J., Shi, X., Fan, Q., & Liu, X. (2022). Toward intent-driven network management: Challenges, use cases and architecture. IEEE Network, 36(2), 60–67. https://doi.org/10.1109/MNET.111.2100162

[22] Chen, M., Herrera, F., & Hwang, K. (2018). Cognitive computing: Architecture, technologies and intelligent applications. IEEE Access, 6, 19774–19783. https://doi.org/10.1109/ACCESS.2018.2791467

[23] Mijumbi, R., Serrat, J., Gorricho, J. L., Bouten, N., De Turck, F., & Boutaba, R. (2016). Network function virtualization: State-of-the-art and research challenges. IEEE Communications Surveys & Tutorials, 18(1), 236–262. https://doi.org/10.1109/COMST.2015.2477041

[24] Yao, H., Mai, T., Wang, J., Ji, Z., Jiang, C., & Qian, Y. (2020). Resource trading in blockchain-based industrial Internet of Things. IEEE Transactions on Industrial Informatics, 15(6), 3602–3609. https://doi.org/10.1109/TII.2019.2903049

[25] Stampa, G., Arias, M., Sanchez-Charles, D., Muntés-Mulero, V., & Cabellos, A. (2017). A deep-reinforcement learning approach for software-defined networking routing optimization. arXiv preprint arXiv:1709.07080. https://arxiv.org/abs/1709.07080

[26] Huang, X., Yu, R., Liu, J., & Shu, L. (2021). Parked vehicle edge computing: Exploiting opportunistic resources for distributed mobile applications. IEEE Access, 7, 14410–14423. https://doi.org/10.1109/ACCESS.2019.2894349

[27] Yu, R., Kilari, V. T., Xue, G., & Yang, D. (2022). Load balancing for interdependent IoT microservices using actor-critic deep reinforcement learning. IEEE Journal on Selected Areas in Communications, 38(6), 1196–1209. https://doi.org/10.1109/JSAC.2020.2986618

[28] Masood, B., & Bhatt, R. (2023). Autonomous network management using deep reinforcement learning: A survey. Journal of Network and Computer Applications, 211, 103563. https://doi.org/10.1016/j.jnca.2023.103563

[29] Azzouni, A., & Pujolle, G. (2020). NeuTM: A neural network-based framework for traffic matrix prediction in SDN. NOMS 2020 IEEE/IFIP Network Operations and Management Symposium, 1–8. https://doi.org/10.1109/NOMS47738.2020.9110403

[30] Cheng, H., Yang, S., & Cao, J. (2022). Dynamic routing in SDN/NFV-enabled satellite-terrestrial networks using deep reinforcement learning. IEEE Transactions on Network and Service Management, 19(3), 2442–2455. https://doi.org/10.1109/TNSM.2022.3152428

[31] Jiang, W. (2022). Graph neural network for traffic forecasting: A survey. Expert Systems with Applications, 207, 117921. https://doi.org/10.1016/j.eswa.2022.117921

[32] Papadimitriou, P., Wang, Z., Primorac, M., Kostic, D., & Raza, S. (2021). Security in SDN/NFV-based networks: Survey and research challenges. Computer Networks, 196, 108218. https://doi.org/10.1016/j.comnet.2021.108218

[33] Musumeci, F., Rottondi, C., Nag, A., Macaluso, I., Zibar, D., Ruffini, M., & Tornatore, M. (2019). An overview on application of machine learning techniques in optical networks. IEEE Communications Surveys & Tutorials, 21(2), 1383–1408. https://doi.org/10.1109/COMST.2018.2880039

[34] Zhou, X., Li, R., Chen, T., & Zhang, H. (2021). Network slicing as an enabler for 5G: Lessons learned from standardization, experimental trials, and live deployments. IEEE Network, 35(6), 282–290. https://doi.org/10.1109/MNET.011.2100063

[35] [35] Luong, N. C., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y. C., & Kim, D. I. (2021). Applications of deep reinforcement learning in communications and networking: A survey. IEEE Communications Surveys & Tutorials, 21(4), 3133–3174. https://doi.org/10.1109/COMST.2019.2916583

[36] Guo, Z., Liu, J., Zhang, W., Wang, J., & Wu, Y. (2023). Reinforcement learning empowered intelligent traffic engineering for multimedia services in software defined networking. IEEE Transactions on Multimedia, 25, 1112–1124. https://doi.org/10.1109/TMM.2021.3131065

[37] Liao, W. H., Kuai, S., & Luo, M. X. (2021). An unsupervised learning-based network state compression approach for QoS routing in SDN. IEEE Access, 9, 6195–6207. https://doi.org/10.1109/ACCESS.2021.3049455

[38] Gebremariam, G. G., Piran, M. J., & Cho, S. (2023). DQN-TE: Deep Q-network based traffic engineering for load balancing in SDN. Computer Communications, 200, 1–12. https://doi.org/10.1016/j.comcom.2022.12.020

[39] da Silva, A. S., Neto, P. H. R., Granville, L. Z., & Rockenbach Tarouco, L. M. (2020). Hybrid reinforcement learning-based traffic engineering in SDN. IEEE Transactions on Network and Service Management, 17(4), 2241–2254. https://doi.org/10.1109/TNSM.2020.3016738

[40] Abdelmoniem, A. M., & Bensaou, B. (2021). Reconciling mice and elephants in data center

networks with deep reinforcement learning. IEEE/ACM Transactions on Networking, 29(4), 1809–1821. https://doi.org/10.1109/TNET.2021.3068979

[41] Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2017). Resource management with deep reinforcement learning. Proceedings of the 15th ACM Workshop on Hot Topics in Networks, 50–56. https://doi.org/10.1145/3005745.3005750

[42] Shi, Y., Zhang, J., O'Brien, D. C., & Letaief, K. B. (2020). Large-scale convex optimization for ultra-dense cloud-RAN. IEEE Wireless Communications, 22(3), 58–65. https://doi.org/10.1109/MWC.2015.7143323

[43] Mai, T., Yao, H., & Zhang, N. (2022). Intelligent network management and control: A survey. Transactions on Emerging Telecommunications Technologies, 33(2), e4253. https://doi.org/10.1002/ett.4253

[44] Zhang, J., Chen, T., & Liu, S. (2024). Federated reinforcement learning for distributed traffic engineering in next-generation SDN. IEEE Transactions on Communications, 72(1), 1–14. https://doi.org/10.1109/TCOMM.2023.3328471

[45] Cui, Y., He, W., Ni, C., Guo, C., & Liu, Z. (2024). Energy-efficient resource allocation in SDN-based IoT networks using deep reinforcement learning. Computer Networks, 240, 110155. https://doi.org/10.1016/j.comnet.2023.110155

Author Profile

Dr. Amit K. Mogal is an academician and researcher in the field of Computer Science with expertise in cloud computing, container orchestration, distributed systems, and emerging technologies. He is associated with the Department of Computer Science and Applications and has actively contributed to teaching, research, curriculum development, and academic administration. He holds a doctoral degree in Computer Science and has published research papers in national and international journals and conferences. His current research interests include cloud-native computing, Kubernetes, container orchestration platforms, virtualization, artificial intelligence, and performance evaluation of distributed systems. Dr. Mogal is committed to promoting quality education, innovation, and outcome-based learning while mentoring students and encouraging research activities.